Faculty of Arts
University of Helsinki

Generating Creative Language - Theories, Practice and Evaluation

Department of Digital Humanities

The Doctoral Programme for Language Studies

Mika Hämäläinen

DOCTORAL DISSERTATION

To be presented for public discussion with the permission of the Faculty of Arts of the University of Helsinki, on the 28th of October, 2020 at 13:00 o'clock. The defence is open for the audience through remote access.

Helsinki 2020

Advisors: Jack Rueter and Jörg Tiedemann
Pre-examiners: Tony Veale and Hugo Gonçalo Oliveira
Opponent: Tony Veale
Faculty representatives: Tuomo Hiippala and Krister Lindén
Custos: Jörg Tiedemann

https://mikakalevi.com
https://rootroo.com

The Faculty of Arts uses the Urkund system (plagiarism recognition) to examine all doctoral dissertations.

ISBN 978-951-51-6706-4 (nid.)
ISBN 978-951-51-6707-1 (PDF)

Unigrafia
Helsinki 2020

Abstract

This thesis presents approaches to computationally creative natural language generation focusing on theoretical foundations, practical solutions and evaluation. I defend that a theoretical definition is crucial for computational creativity and that the practical solution must closely follow the theoretical definition. Finally, evaluation must be based on the underlying theory and what was actually modelled in the practical solution.

A theoretical void in the existing theoretical work on computational creativity is identified. The existing theories do not explicitly take into account the communicative nature of natural language. Therefore, a new theoretical framework is elaborated that identifies how computational creativity can take place in a setting that has a clear communicative goal. This introduces a communicative-creative trade off that sets limits to creativity in such a communicative context. My framework divides creativity in three categories: message creativity, contextual creativity and communicative creativity. Any computationally creative NLG approach not taking communicativity into account is called mere surface generation.

I propose a novel master-apprentice approach for creative language generation. The approach consists of a genetic algorithm, the fitness functions of which correspond to different parameters defined as important for the creative task in question from a theoretical perspective. The output of the genetic algorithm together with possible human authored data are used to train the apprentice, which is a sequence-to-sequence neural network model. The role of the apprentice in the system is to approximate creative autonomy.

Evaluation is approached from three different perspectives in this work: ad-hoc and abstract, theory-based and abstract, and theory-based and concrete. The first perspective is the most common one in the current literature and its shortcomings are demonstrated and discussed. This starts a gradual shift towards more meaningful evaluation by first using proper theories to define the task being modelled and finally reducing the room for subjective interpretation by suggesting the use of concrete evaluation questions.

In Memoriam

Respecting the memory of my mother Irma Hämäläinen, who passed away when I was 14 years old, my dog Lunni, who passed away at the age of 12, and my sister Mari Latto, who passed away last year.

Contents

List of original publications

This thesis is based on the following publications:

I Hämäläinen, M., & Alnajjar, K. (2019). Generating Modern Poetry Automatically in Finnish. In *2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing: Proceedings of the Conference* (pp. 6001–6006). Stroudsburg: The Association for Computational Linguistics.

II Hämäläinen, M., & Alnajjar, K. (2019). Modelling the Socialization of Creative Agents in a Master-Apprentice Setting: The Case of Movie Title Puns. In *Proceedings of the 10th International Conference on Computational Creativity* (pp. 266-273).

III Hämäläinen, M., & Alnajjar, K. (2019). Let's FACE it: Finnish Poetry Generation with Aesthetics and Framing. In *12th International Conference on Natural Language Generation: Proceedings of the Conference* (pp. 290-300). Stroudsburg, PA: The Association for Computational Linguistics.

IV Hämäläinen, M., & Honkela, T. (2019). Co-Operation as an Asymmetric Form of Human-Computer Creativity. Case: Peace Machine. In *Proceedings of the First Workshop on NLP for Conversational AI* (pp. 42–50). Stroudsburg: The Association for Computational Linguistics.

V Hämäläinen, M., Partanen, N., Alnajjar, K., Rueter J. & Poibeau T. (2020). Automatic Dialect Adaptation in Finnish and its Effect on Perceived Creativity. In *Proceedings of the 11th International Conference on Computational Creativity*. (pp. 204-211)

VI Hämäläinen, M., & Rueter, J. (2018). Development of an Open Source Natural Language Generation Tool for Finnish. In *Proceedings of the Fourth International Workshop on Computational Linguistics of Uralic Languages* (pp. 51-58). Stroudsburg: The Association for Computational Linguistics.

The publications are referred to in the text by their roman numerals.

Abbreviations

BRNN	bi-directional recurrent neural network
CC	computational creativity
E2E	end-to-end
FST	finite state transducer
GA	genetic algorithm
GAN	generative adversarial network
IPA	international phonetic alphabet
LSTM	long short-term memory
NLG	natural language generation
NLP	natural language processing
NMT	neural machine translation
RNN	recurrent neural network
seq2seq	sequence-to-sequence

1. INTRODUCTION

Creativity has puzzled philosophers from the times immemorial. Unsurprisingly, it has provoked a whole deal of theoretical work aiming to capture what the phenomenon is really about dating all the way back to Plato (see Asmis, 1992). In fact, Gaut (2012) divides philosophical takes on creativity into two categories: irrationalism and rationalism. Plato belongs to the former category, which assimilates creativity with madness or intuition. However, Plato would consider art requiring craftsmanship rational. Rationalism highlights creativity as an act requiring reasoning such as planning a plot so that it elicits certain emotions in the reader.

Moving away from the previous dichotomy, one of the most cited theories on human creativity in the computational contexts is the four Ps by Rhodes (1961). According to this view, creativity consists of person, process, product and press. Person refers to the psychological traits and capabilities such as intelligence, attitudes and values of a creative individual. Process highlights the importance of how something is created rather than focusing only on what is created, namely the product. Press describes the wider sociocultural context, or environment, where creativity takes place.

Most importantly, creativity is something that we have considered fundamentally human (cf. Hennesey & Amabile, 2010) for a long time in the history of mankind. Yet, what is considered creative, is inherently socially constructed. For instance, Shao et al. (2019) report that usefulness is considered more important for creativity in the East, while in the West novelty is considered an equally or more important attribute of creativity. There seems to be a tendency for people to think that creativity should only occur in the biological. Pease & Colton (2011) have called this ideology carbon fascism based on the critique on machine creativity expressed by Bedworth & Norwood (1999).

This thesis presents my research on creative natural language generation (NLG). This research is conducted within the paradigm of computational creativity. Colton and Wiggins (2012) characterize the discipline as, "the philosophy, science and engineering of computational systems which, by taking on particular responsibilities, exhibit behaviours that unbiased observers would deem to be creative", a definition which has later become one of the most fundamental ones in the field.

The establishment of computational creativity as a field of research took place over a decade ago, and this year the major conference dedicated solely on the field,

ICCC (International Conference on Computational Creativity) reaches its 11th annual iteration. Furthermore, the first call for papers of the Journal of Computational Creativity was announced as recently as last year. Despite this relatively long time frame, I have identified two major drawbacks that have not been solved to a satisfying degree. Firstly, there is a plethora of practical work on generative systems and a great deal of theoretical papers describing computational creativity. There is, however, a gap between the two; practice is hardly ever motivated by theories. Secondly, evaluation has always been the Achilles' heel of the research conducted on the field (see Lamb et al. 2018).

In this thesis, I will further elaborate on the two problems and my take on advancing the state of affairs with respect to both of them. I will show results from following the current practices (paper I) and present two different creative systems based on two different theoretical points of view (papers II and III). I will show the shortcomings of the current evaluation methods (paper V). Furthermore, I will present a theoretical framework of my own to fill a theoretical void in the field (paper IV). In order to solve NLG challenges for Finnish on a practical level, I have authored an open-source library (paper VI).

The major technical contribution of my dissertation is the introduction of the master-apprentice approach, which joins the interpretability of a traditional machine learning approach, namely that of a genetic algorithm (GA), with the flexibility and adaptability of a neural network. What makes the GA interpretable is that it uses several different fitness functions each of which measures a different desired attribute. Both of these approaches serve a purpose for the overall system. In this thesis, I will argue for interpretability of the aesthetics the system follows in order for it to satisfy the requirements set for computational creativity. However, such a system is oftentimes rather static and cannot be autonomous in its creativity, hence I combine it with neural networks that can learn to adjust their standards more freely based on the training data.

Whereas my work methodologically follows the similar empirical machine learning as a great majority of current NLP research does, I would like to draw more attention into the question of evaluation throughout my thesis. In a field that relies heavily on automated methods, human evaluation is often conducted in an ad-hoc fashion without questioning what such an evaluation can or is supposed to reveal about a given computational system. Therefore, I propose more reasoned ways of approaching the problem of evaluation, especially in papers II and III.

2. Theories of Computational Creativity

In the field of computational creativity, a myriad of different theories has emerged that try to identify what computational creativity exactly is or should be. The theories focus on creativity from a computational perspective, rather than the perspective of what creativity means when exhibited by people.

I myself do not agree with the argument that computational creativity should not be the same as human creativity. However, it is true that with the current state of computational models, computers cannot be creative in the exact same fashion as real people, but this is, by no means, a maxim of computational creativity. What I do agree with, is that we should not approach computational creativity from the same theoretical perspective as we approach human creativity. Mostly due to the fact that a lot is still unknown about human cognition and how creativity actually emerges in us, despite extensive theoretical work, whereas how we are modelling creativity in machines is very much known to whoever seeks to build a creative system.

Computational creativity research has tried to deviate itself from any research focusing on generation, such as regular NLG research. A pejorative term *mere generation* has been used in the field pointing out that systems that are not creative, are just *merely generating*. However, the term itself could be better defined and it seems to mean different things for different authors, and the most authoritative and cited source for the term is a blog post by Veale[1] that is unavailable at the time of writing in the original url.

What I understand by *mere generation* and how I would define it is that the system does not operate on any definition of creativity. As I have already pointed out in the introduction, creativity is an inherently socially constructed concept. It means different things to different people at different times. For anyone to argue that their system is creative, they must also state what they mean by creativity itself. For as long as no definition is provided, anything goes for as long as it produces output convincing enough to fool people into seeing more than what the intention of the system ever was (see. 5. On Evaluation of Computationally Creative Systems for more)

---

1 https://web.archive.org/web/20170402185912/http://prosecco-network.eu/blog/scoffing-mere-generation

If a definition for creativity is provided, the problem becomes easier to model, and it becomes easier to critically assess the degree to which a system is creative (see Alnajjar & Hämäläinen, 2018). This thinking is at the core of my thesis and it is reflected by papers II and III. Mainly, the key idea is based on the SPECS approach (Jordanous, 2012), which states that creativity must be first defined on an abstract level, and then on a concrete level. Finally, the evaluation of the system should be in line with the initial definitions.

I do agree with the ideas presented in the SPECS approach, but I find the definition for creativity proposed in the paper troublesome (see 2.5. SPECS for more). Furthermore, I think that an additional requirement for the SPECS ideology is needed; that is an alignment between the definition, implementation and evaluation. One can have an apt definition and a suitable evaluation for the definition, but if the implementation does not even try to reflect what has been defined, the evaluation results are hardly representative of the creativity of the system (see 5. On Evaluation of Computationally Creative Systems for more elaboration on this thought).

The remainder of this section is dedicated to presenting some of the existing, influential theories on computational creativity. The most crucial sections for my thesis are sections 2.3.-2.5. as they serve as the theoretical foundation of my work in papers II and III, although some of the ideas of Section 2.1. are present in paper II and the overall ideology presented in Section 2.2. is remotely touched in paper I. Section 2.7. Asymmetric Creativity and Conveying a Message is dedicated to describing my theoretical work on filling a theoretical void in the existing computational creativity theories as described more profoundly in paper IV.

Creative autonomy, as defined by Jennings (2010), requires three components for a system to be considered autonomous in its creativity. The system should be able to evaluate its own creations without external judgment, change its own standards without being explicitly told how to do so, and none of the first two requirements may rely on pure randomness. The work presented in papers II and III approximates this notion with the master-apprentice framework.

2.1. Boden on Creativity

There is no doubt that Boden (2004) is one of the most celebrated takes on what it means to be creative within the computational creativity paradigm. Two ideas of hers are raised time after time in computational creativity research. The first is a simple dichotomy that divides creativity into two categories according to the novelty value: P- and H-creativity.

P-creativity, or psychological creativity also known as personal creativity, is any kind of creative innovation that is only novel to the creator itself. Essentially this means that if a person figures out how to solve a puzzle, such as a Rubik's cube, they are exhibiting P-creative behavior as many people before them have solved the same puzzle in the past and many people will continue to do so in the future.

If the novelty value reaches to a more global context, we are dealing with H-creativity, which is human or historical creativity. This means creativity that is a game changer in the way of thinking and is unlikely to occur often by different people. An example of this is the use of fire which revolutionized the history of mankind.

In paper II, I am, however, assimilating H-creativity with a more down to earth and easier to achieve creative behavior. Mainly due to the fact that the paper deals with humor generation and a joke, no matter how good, can hardly be contrasted to an achievement of the nature of the fire. Therefore the H-creative is seen more as something non-obvious i.e. something people would not come up with easily by themselves or have not heard before.

The other set of important notions Boden introduced have to do with how the creative process unfolds. These are *exploratory, transformational* and *combinational creativity*. The latter of which is often not seen as important as the first two, and is for instance, omitted in the later formalization of Boden's three-fold definition (see 2.2. Formalization of Boden's Theories). Combinational creativity is related to coming up with something new by combining familiar concepts. A good example would be an analogy, which can be used to explain something unfamiliar with familiar terms.

Exploratory creativity is related to the idea that there is a conceptual space that consists of creative artifacts. Exploratory creativity is similar to conducting a search in a search space (=conceptual space). The search space can consist of complete or partial solutions. In this sense, the poems one can create are all found within the search space where creative thought occurs. Nothing outside of the space can be created.

As exploratory creativity is clearly limited by the bounds of the search space, there are only so many things one can create. A change in the rules that define the search space introduces a paradigm shift. This is known as transformational creativity as it transforms/changes the space where creativity occurs and thus enables us to come up with completely new solutions that were not available in the original search space, such as completely different kinds of poems.

I have intentionally used the word *search space* here instead of the original wording *conceptual space* to draw a closer link between these notions and genetic algorithms (GA), a model that we have employed in papers I, II and III. I will provide a more detailed link between the GA and these notions in the following section.

2.2. Formalization of Boden's Theories

In order to better situate Boden's concepts of exploratory and transformational creativity in the field of computational creativity, Wiggins (2006) has presented his formalization and extension on the two notions. He begins his argumentation by introducing the concept of a universe. The universe contains everything possible, from the desired creative artefacts to artefacts of a completely different nature. A conceptual space is then defined within the universe for conducting creative search. Such a space is selected by an interpretation function by applying rules that define the space. For example, a context-free grammar might be such an interpretation function that sets the limits to what can be generated according to its rules. In other words. it can never generate sentences that are not covered by its rules.

In addition to rules defining the search space, Wiggins identifies another set of rules that are used to traverse it. A context free grammar can be run to generate all the possible sentences, or the search space can be traversed in a more informed fashion by following heuristics of some sort.

When the system conducts search, or traverses the conceptual space, it needs an evaluation function to assess the quality of the concepts it finds. This is important so that the system can know when it has found a creative artefact of a value, as the search space is bound to contain many bad solutions. These are the requirements for exploratory creativity.

For transformational creativity, the system needs to be able to change the way in which it conducts the search. This can be done in two ways, either by changing the rules that define the search space or by changing the rules that are used to traverse it. While Wiggins states the possibility of changing the evaluation functions as well, he does not describe that possibility extensively.

In terms of the GA, there is quite an overlap in these theoretical notions. Without going too deep on the technicalities that are to follow in Section 3.3. The Master-Apprentice Approach, GA conducts search by mutating and crossing individuals over with each other. At the end of each generation, the individuals are evaluated, and the fittest ones are left to survive for the next generation. The possible

search space is defined by the initial individuals and how the mutation and crossover are implemented. In our master-apprentice framework, the transformational creativity is left for the apprentice, as the search space can change depending on the training data used. Different variations of the training data are discussed in papers II and III.

2.3. The Creative Tripod

Colton (2008) presents a computational creativity framework called the creative tripod. The tripod has three legs: *skill*, *imagination* and *appreciation*. Creativity cannot occur without all three legs being present simultaneously. This theory is important for my thesis as it is the foundation of the work presented in paper II.

Skill means the capability of performing the creative action, producing an artefact. A painter will have multiple techniques in his arsenal, or a sculptor will know how to shape a statue out of marbel. Skill itself does not yet entail creativity; if a highly skilled painter copies an existing painting with the highest attention to detail, he still does not achieve creativity according to the tripod.

Imagination refers to the fact that creativity is not supposed to be repetitive. A degree of novelty is needed in the created artefacts. A skillful painter painting the same painting over and over again is hardly creative. I would like to point out the similarity of the concept of imagination with Boden's P- and H-creativity, both of which essentially relate to the novelty of the created artefact. This assimilation is also presented in paper II.

Appreciation is the last leg of the tripod. It simply refers to the capability of assessing one's own creations. A generative system can easily produce a lot of artefacts (high imagination) based on a skill learned from data. However, such a system is still not creative unless it can appreciate its own output. Furthermore, in paper II, I argue that appreciation should be nuanced reflecting the individual attributes that constitute humor rather than a single probabilistic value. I would also like to point out the similarity with the evaluation function described in Section 2.2. Formalization of Boden's Theories.

The creative tripod identifies that computational creativity is a joint effort between the programmer, the computer and the consumer, each of which can bring their effort to the overall creativity of the system. My view on this, which I also hold in paper II, is that the system must exhibit all of the legs of the tripod. Nonetheless, the degree to which they come from the programmer is another question. In paper II, the master inarguably exhibits all the different legs of the creative tripod to the

extent they were defined to produce humorous headlines. This is due to the fact that the fitness functions and how they operate has been defined by us, the programmers. The situation gets more difficult for the apprentice, which is not defined by a programmer per se, but the data it learns from. In such a case, the presence of a programmer is considerably smaller, but the degree to which appreciation is exhibited by the system becomes much less clear. For the future, I would propose an extension to the creative tripod framework taking the training data into account as well as one of the contributing factors to the creativity of the system. This theoretical extension will allow for a better inspection of the effect of the training data.

It is inevitable that the consumer is an important party in the creative act of the system. Computational creativity is typically interested in producing pleasing, artistic artefacts, which are meant to be consumed by people. However, any system that completely outsources any of the legs of the creative tripod to the consumer, cannot be creative in my view. This point will become evident in Section 5. On Evaluation of Computationally Creative Systems.

2.4. FACE

FACE (Colton et al, 2011) identifies four components that should be present in a computationally creative system: *framing*, *aesthetics*, *concepts* and *expression*. All of these can be characterized on the ground level and process level i.e. what they are and how they came to be. This theory is particularly important for paper III, as it serves as its theoretical basis. I will describe these components in an inverse order from the most concrete to the most abstract.

On the ground level, expressions are the output, the artefacts created by the system. This simply means artefacts such as poetry or humor. On the process level, expressions are the output for a given input to a concept.

Concepts are easiest to understand as programs, for on the ground level, the term refers to the creative program that produces expressions. On the process level, the question becomes how the program was made. This is fundamentally related to my earlier discussion in Section 2.3. The Creative Tripod, that is, how much of the concept is due to the efforts of the programmer and how much due to the data used to train the system.

Aesthetics is close to the evaluation function of Wiggins (2006) and appreciation of the creative tripod (Colton, 2008). Again, the system should be able to assess the aesthetic value of its own creations on the ground level. On the process level, a

similar question raises as in the case of concepts. What is the degree to which the aesthetic measures are defined by the programmer and by the data.

The last notion of the theory, framing, is what sets it clearly apart from other theories on computational creativity. Framing is a co-text that is to be presented together with the creative artefact, such as a description of the author's life or a similar side note typically seen in art museums next to a painting. Framing can be used for multiple purposes such as for persuading people into believing in creativity of the system by providing more historical or cultural context. The use of framing for deception is also discussed in Cook et al. (2019) among other uses. In our work presented in paper III, I reject the use of framing for such a purpose. The way framing is used there is to expose the internal aesthetic measures for the purpose of human evaluation. Therefore, the use of framing is only to provide the information the system had during the creative process, not to come up with an insincere background story or interpretation.

It is important to note that one thing that is usually associated with creativity, namely novelty, is completely missing from this framework. I state this to further highlight how differently creativity can be understood and defined, even by contemporary theories.

2.5. SPECS

The main ideology of the SPECS approach (Jordanous, 2012) was already discussed in the introductory text of this main section, namely that the approach requires creativity to be defined on an abstract level, then on a concrete level reflecting the particular creative task in question, and finally the evaluation of the system should be based on the definitions. However, the approach comes with its own, preferred, definition for creativity. Here I will describe it and explain why it is not adequate for my main argument of an alignment between theory, implementation and evaluation. SPECS lists altogether 14 key points for creativity.

A creative system should have a deliberate effect on the creative process and it should be able to cope with uncertainty and incomplete answers. There should be domain competence and general intellect present in the system. I find the term general intellect especially challenging as a starting point for a definition of a creative system. If defining creativity is a challenging task, using intellect as one of the terms to define it, opens up another equally difficult to define concept. In fact, there is a lot of debate in the field of artificial intelligence about what it really means to be artificially intelligent (see McCharty, 2007).

According to SPECS, the system should be able to generate results or reach a goal in a novel way, and do so in an independent way. There should be personal or emotional involvement or intention or desire to perform a task. This requirement, again, is one that is very difficult to model computationally. In fact, there is an entire field of science dedicated to modelling emotions computationally with a multitude of different theoretical starting points (see Marsella et al., 2010).

Furthermore, the SPECS states that there should be progression and development in the process and creativity should take place in a social environment. Subconscious processing and thinking as well as evaluation are also mentioned as important factors of creativity. However, again, in order to argue for subconsciousness or thinking in a creative system, one first has to deal with the problem of computational consciousness. The problem is that currently, there is no agreement on the nature of human consciousness either, whether it simply emerges from a complex system or is a feature of the biological brain fully explainable by the firing of the neurons (see Kim, 2018).

Finally, the approach states that there should be value and variety in the output. The main reason I do not think that this 14 point definition for creativity can be used as a starting point for any computationally creative system is that it includes terminology that is fully human (such as emotions, consciousness, intelligence) and the computational modelling of which is a challenge equally difficult to creativity, if not more so. Especially if one seeks to have the problem definition, implementation and evaluation in line, implementing and evaluating these notions becomes difficult if not impossible.

2.6. Co-Creativity

In this section, I will briefly describe some of the theories on co-creativity, that is a scenario where a human is engaging with a creative system. This section serves mainly as a background for the following section, where I present my own theoretical extension to this line of theoretical work. Lubart (2005) identifies four different scenarios for co-creativity where a computer acts as a: nanny, pen-pal, coach or colleague.

In a nanny situation, the computer sets deadlines and guides the user in a creative task. This does not require too much creativity from the computer, but rather functionalities related to keeping the user's work effective and organized. In a pen-pal scenario, the computer is more of a platform that enables the exchange of creative ideas. Again, such a system does not need to exhibit any creative behavior

of its own. A computer acting as a coach can help the user in engaging in creative thinking by displaying information in novel ways. In this situation, the system does not need to exhibit creativity on its own, but it certainly needs to know about creativity in order to promote and propagate more creative thinking. Finally, if the computer acts as a colleague, it is in a real partnership with the human user, as a creative equal.

Davis (2013) introduces the concept of human-computer co-creativity that aims to bring HCI and computational creativity closer to each other. This notion adheres to the ideology of having a computer as a colleague in the creative process. In this view, the computer should be able to adapt and react to the input provided by the user in creative and novel ways. The interaction sequence can be unpredictable and does not need to follow a specific script, which makes the task of coping with such input more demanding for the machine.

Mixed-initiative co-creativity, a notion elaborated by Yannakakis et al. (2014) is also a manifestation of a computer as a colleague. The important aspect is that the two parties, the human and the computer, can actively participate in the creative behavior, but not necessarily to the same extent. They demonstrate this theory in the context of a game designing system.

2.7. Asymmetric Creativity and Conveying a Message

In this section, I present the overall theoretical contribution of paper IV. The existing theoretical work on co-creativity highlights that there is always human creativity present, and computational creativity may or may not be present, on the one hand. On the other hand, computational creativity theories usually highlight aesthetics or value of the output ignoring the most fundamental function of human language; communication.

We humans hardly ever sit on a chair in a corner of a room and generate aesthetically pleasing utterances with no understanding or intention behind the message we want to convey with our words. An exception to this might, of course, be a patient suffering from Wernicke's aphasia, but in the broad sense, we usually verbalize something we want to communicate. In my work, I have called any effort of generating language without a message, a meaning to convey, mere surface generation in the spirit of mere generation. I am not claiming that my practical work (papers I, II or III) or any other work on computational creativity I am aware of is trying to be anything beyond mere surface generation. Nevertheless, it is important to subject this notion to theoretical inspection.

Based on the discussions I had when presenting this paper, I feel the need to clarify what I mean by conveying a message. For instance, Loller-Andersen & Gambäck (2018) present a system that can produce poetry about an image given to the system. The argument against mine was that the system does have a message, that is the content of the image. Now, going back to my previous example of a person sitting in a corner and uttering pleasing sentences, even if these sentences were about an image and somewhat related to it semantically, it still does not entail any message in particular; anything even remotely related fills the purpose. For instance if we say, "it is raining outside", we probably want to communicate the meaning of a particular kind of weather occuring outdoors rather than communicating something about a picture of an umbrella. It is not uncommon to have an input for a creative system that is used to generate output related to it (c.f. Veale & Alnajjar 2016; Gervás, 2018; van Stegeren & Theune, 2019). For my theoretical work, however, communicating about something is not the same as communicating something.

As it is difficult to grasp what it would even mean for a computer to have a message that it needs to communicate, especially in very artistic text generation, such as poetry, stories or movie scripts, without resorting to an unnecessary degree of anthropomorphism, I have decided to take goal-oriented dialog under my theoretical inspection. In such a system, even though there is no internal motivation for a computer to communicate its inner thoughts (whatever those might be), there is typically some information content that needs to be communicated, such as the timetable for a train or the price of plane tickets. Although there is a lot of practical work done on dialog generation/adaptation relating to the field of computational creativity (c.f. Shen et al. 2017; Wei et al. 2018; Hämäläinen & Alnajjar, 2019; Chikai et al., 2019), the approaches usually fail to demonstrate that the intended message (if any) is conveyed as intended and that there is creativity in the process.

In the paper, I commence describing my theoretical framework by presenting existing related theories in an interdisciplinary fashion. Moreover, I identify the main limiting factors, that is the maxims of the co-operative principle (Grice, 1975), as the core of the framework. In order to ensure that the goal oriented dialog system still excels at conveying its message effectively, it should follow the co-operative maxims: manner, quality, quantity and relevance. I incorporate other theoretical notions and their interdependencies to the model reaching a framework that sets the limits the system has to fulfill regardless of its supposed creativity.

I identify a communicative-creative trade off. If a system optimizes communication of the desired message as purely and fully as it is required by the

context, only very little room is left for creativity. In contrast, if the system goes to the other extreme of creativity, it will probably try to communicate the message very creatively, for example, as a riddle, which jeopardizes the understandability of the message in the recipient. Knowing the limits to how creative and how communicative the system needs to be is contextually dependent and requires understanding of the user, a user-model in other words. Thus a balance between creativity and predictability must be maintained.

In order to better know how creativity can manifest itself in a conversation governed by strict linguistic, cognitive and social rules, I identify three types of creativity that can occur in a conversational setting: message creativity, contextual creativity and communicative creativity.

Message creativity can occur in the level of altering the denotation of the message. For instance, the expressions *a glass is half empty* or *half full* communicate essentially about the same phenomenon, yet their denotations are very different. A system can purposefully alter what it sets to communicate for as long as the same idea gets through. It is possible for the system to seek to alter the connotation of the message as well. If the context allows, the system can opt for a formal or for a more casual wording of the message delivering different connotation, for instance, about social distance or emotional affect. Finally, the system can purposefully exploit the speech acts (see Searle, 1969), for example in order to communicate an expressive message with a directive surface form.

The notion of contextual creativity gives a lot of room for creative exploration of the context. It partly relies on a user model that can enable computational approaches to intersubjectivity, namely if the system knows the user well enough, it can predict whether certain devices of figurative language, such as metaphors or sarcasm, will be understood as intended by the user. As we typically take a role in conversation (see Goffman, 1959), a system can actively try to shift its role for something that enables more creativity if the context permits that. Finally, the system can take the time perspective of the conversation into account and either conduct creative planning of the future directions of the conversation or bring up past events of the conversation in a new light.

Communicative creativity can occur by selecting a new social script for the conversation that allows for more creativity. It is also important to adjust the level of elegance that the system aims for in the conversation. Elegance is understood as the most minimalistic way possible for conveying a message as fully as possible. In conversation, aiming for minimalism might rule out a lot of the creative potential that the system could otherwise unlock. Finally, the system can also deviate from the

co-operative maxims in an informed way. The system can communicate more than what is necessary, for instance, if the communicative output still contributes to the conversation.

The theory is asymmetric in the sense that it does not expect any creativity to be present in the human user, but only in the computational system. The paper includes an adaptation of the theoretical model to the Peace Machine thinking as elaborated by Honkela (2017). The main focus of the Peace Machine is to challenge AI for social good, and especially in contributing to the process of peace by enabling creative thought, meaning negotiation and enhanced intersubjectivity. The goals of the system are, however, beyond the scope of my dissertation.

3. Practical Applications

This section is dedicated to describing the existing practical work on the two main tasks of computational creativity presented in papers I-III. In addition, I present the practical solutions taken in my papers in this section. I do not yet take any stance on the theoretical aspect of my work for how the theories and practice intertwine is presented in section *4. From Theories to Practice.*

3.1. Related Work on Humor Generation

Humor generation, and pun generation in particular, has received quite some attention in recent years. Automated humor detection has also provoked some research interest (Yang et al., 2015; Bertero et al., 2016; Chen & Soo, 2018) as more and more annotated data have become available such as the recent Humicroedit dataset from last year (Hossain et al., 2019). However, as humor detection typically operates on a particular dataset learning the particular humor in that specific dataset, it is quite a distant task from generation, especially since generation of the creative kind is supposed to produce something novel and unexpected as well. Although detection/appreciation is a key part of creative generation, I will not describe approaches dedicated purely to detection in this section as their starting point is typically different from that of generation. In this section, I will focus more on the recent humor generation systems, for an overview across a longer timespan, see paper II.

Humor generation has been approached from two different disciplines during the past few years. On the one hand there are the researchers within the computational creativity (CC) paradigm who try to model creativity or at least novelty in their approaches, on the other hand, there are people in the field of NLP who have proposed systems from a merely generative standpoint. I will start by describing some of the recent approaches arising from the field of NLP before moving to the discourse taking place in CC.

Aggarwal & Mamidi (2017) propose a system generating *Dur se dekha* jokes in Hindi. These jokes follow a predefined structure similar to knock-knock jokes in English. Their method is based on hand written templates and a manually collected

lexicon. They use three constraints to pick a suitable punchline from the lexicon: semantic category, grammatical gender and form constraint expressed by rhyming or Levenshtein distance. They evaluated their system on a 5-point Likert scale by consulting 15 people, and their system achieved nearly human-level in naturalness (3.16 vs 3.33), but scored only mildly on the humor aspect (2.4).

A deep learning take on homographic pun generation was proposed by Yu et al. (2018). They use a conditional LSTM based language model to generate a punny sentence that has both of the desired senses for an input word present at the same time. They use the English Wikipedia with word sense disambiguation labels predicted by an existing tool. They evaluate their system by consulting only 5 people on Amazon Mechanical Turk. Their system achieved results that are considerably lower than human authored puns but higher than their baseline system on the abstract scale of fluency, accuracy and readability.

Luo et al. (2019) present a method for generating polysemous puns by using a GAN approach. Their model consists of a generator that can produce a punny sentence containing the desired word in two senses, and a discriminator that can predict the real and punny senses of the output of the generator. The generator is trained on Wikipedia data that is automatically tagged with word-sense disambiguation tags. They evaluate their approach using only three annotators. The results of their model were clearly preferred over puns generated by an existing model (57 times vs 24 times), however, human authored puns were preferred almost all the time to the puns generated by their system (79 times vs 8 times).

He et al. (2019) present a system for generating homophonic puns based on a local-global surprise principle. They argue that a pun word should be surprising in a local context and congruent in the global context. In other words, the punny word should seem like a misfit based on the semantics of words that are in its immediate neighbourhood, but still make sense in the level of the whole text. They use a language model to assess the surprise introduced by a punny substitute word. Their system takes in an existing sentence, replaces a word with a punny one and modifies the sentence to support the meaning of the punny word even further. They report that as few as 31% of the puns generated by their system were rated as puns by people on Amazon Mechanical Turk.

The recent work conducted within the computational creativity domain has focused on a variety of different types of humor. Whereas the NLP research is mainly interested in purely ML driven E2E solutions, the CC field finds more use in templates and human intuition when designing the system.

Humor generation in the context of memes in Portuguese has been studied by Gonçalo Oliveira et al. (2016). The system turns headlines into memes by picking a suitable meme image and adapting the headline to a meme like format. For mapping a headline to a meme image, a rule based classifier is used. Headlines are adapted to meme text by either finding a similar proverb based on semantics or rhyming. Predefined templates are then used to render the final meme text. They evaluated the system on a 5-point Likert scale based on coherence, suitability, surprise and humor by altogether 52 people, on the average the score for coherence was 3.81, while for the other 3 metrics the average was closer to 3 (2.98, 3.06 and 3.10 respectively).

A system producing sarcasm that can potentially exhibit humor is presented by Veale (2018). The presented system tries to produce a failure in expectation contradicting a salient property of a concept for sarcastic effect; this is done by consulting existing semantic databases that link concepts to their properties, adjectives to nouns in other words. The system exploits carefully designed templates and rules in sarcasm generation. The paper argues that systems designed for analysis cannot simply be reversed to generate, as an analyzer can yield results due to over-fitting without truly generalizing to understand the phenomenon. The paper presents a study of perceived positivity of the focal word by human judges, but it does not present any evaluation of the full sarcastic sentences.

Winters et al. (2019) proposes a system that generates humor based on schemas that tell the system how the joke should be generated by defining the generator function, template, metrics, aggregator and keywords. The metrics used for humor generation are obviousness of a word (its corpus frequency), conflict in the punchline (n-gram probability), word compatibility (n-gram probability as well) and inappropriateness (corpus frequency). They evaluate the system on a star rating with 203 people 11.41% of the jokes generated by their system got 4 or more stars as opposed to 21.08% for human authored jokes.

The work proposed in paper II does not only differ from the existing work based on the technical approach and theoretical foundation embraced, but also the problem setting is very different. While there is existing work on pun generation, our approach does not rely on polysemy or homonymy (complete or partial), but rather takes a more flexible approach in sound similarity. In addition, the task is more specific as our system does not reach a satisfactory output by producing any sentence containing a desired pun, but rather it has to produce a recognizable movie title delivering a pun.

3.2. Related Work on Poem Generation

Poem generation has a long tradition in computational creativity research (see Gonçalo Oliveira, 2017). Recently, it has also started to gather more interest in the NLP community, most notably by Chinese NLP researchers. In this section, I will first describe some of the recent work conducted on the field of NLP before describing more computational creativity oriented research. It is to be noted that NLP research on poetry does not limit only to generation, but also covers poem analysis (see Kao & Jurafsky 2012; Caccavale & Søgaard, 2019; Rahgozar & Inkpen, 2019).

Zhang et al. (2017) propose a memory augmented neural network model, which, according to their claims, produces more innovative poems than previous neural approaches to Chinese poetry generation. They train an RNN based model with attention to generate poems from input topic words on corpora of Chinese poetry. The output of the system is evaluated by 34 experts based on compliance (rhymes and tones), fluency, aesthetic innovation and scenario consistency on a 5-point Likert scale. From the different models they experimented with, none was clearly the best on all the parameters, but the highest individual average scores were 4.1, 3.01, 3.07 and 3.17 respectively. They do not report the results for theme consistency per model although they stated it as an evaluation criterion.

Lau et al. (2018) model sonnet meter and rhyme as a joint neural model. They train their model on automatically tagged data. The model consists of three LSTM models, a language model, an iambic meter model and a rhyme model, all of which are trained together in a multi-task learning setting. In their human evaluation, they removed the workers who did not perform well enough in control questions. The evaluation task was to distinguish a human written poem from a computer written one. Their best model was indistinguishable from human written poetry 46.8% of the time. However, their expert evaluation suggested that their model was only good at rhyming and meter, while falling short on readability and emotion.

Reinforcement learning has been proposed by Yi et al. (2018). They use the criteria previously used in human evaluation fluency, coherence, meaningfulness and overall quality as reward functions for the algorithm. They train two reinforcement learning models that also learn from each other. In their human evaluation, they recruit 12 experts so that each poem will get 3 different evaluations. They report relatively high scores (between 3.6 and 4.06 on the average) for their best model.

Yang et al. (2019) model poem generation as an unsupervised machine translation problem. Their system takes text written in vernacular Chinese as input and produces a poem in classical Chinese as output. They use an existing unsupervised machine translation approach that consists of an encoder-decoder architecture. They have 30 people evaluate their system based on fluency, semantic coherence, semantic preservability (how much of the meaning of the vernacular Chinese text is preserved in the translation) and poeticness. They use a 5-point Likert scale, resulting in average scores higher than 2 but lower than 2.7 for their best dataset. The approach merely paraphrases poetry from one domain to another and it does not generate novel outputs at each run.

In more of the computational creativity side of research, Manurung et al. (2012) propose a genetic algorithm approach to generating poetry. In their approach, they define that a system must satisfy three criteria for poems; grammaticality, meaningfulness and poeticness. Grammaticality is set as an absolute constraint in the GA. They use a derivation tree grammar formalism as a basis for the generated poetry, the trees are mutated by adding or removing a random subtree during the genetic process. Subtrees may also be swapped by the mutation or crossover. Poeticness is evaluated by comparing the meter of the individual poems to a target meter and meaningfulness is evaluated by semantic similarity to the target semantics. They only conduct evaluation by automated metrics rather than consulting human evaluators.

Gonçalo Oliveira & Alves (2016) present an approach for generating poetry from text by first extracting a conceptual map from the input and then adapting the semantic network used in generation to the extracted concepts. The generator itself consists of multiple modules that have been designed to do different tasks such as line generation or organizing lines in the poem. They elaborate a specific grammar formalism to convert the conceptual map to a form suitable for poem generation. The system can rank its generation candidates based on rhyme and meter. The authors do not present any automatic nor human evaluation of their system.

An n-gram based method has been elaborated by Gervás (2017) to produce poetry that caters for thematic consistency and enjambment in a template free fashion. The paper highlights the problem that existing work usually focuses on a particular aspect of poetry and solves that from an engineering point of view, while this focus is never explicitly stated and the work itself gives the impression as though the much larger problem of good poetry generation was solved to a greater degree. Therefore, this particular paper makes a narrow focus admitting that any features beyond those explicitly solved (thematic consistency and enjambment) are

solely due to serendipity and not to the system's internal capabilities. I completely agree with this argumentation and papers II and III hold a similar stance towards CC. The approach by Gervás (2017) uses a corpus to determine word relatedness and vocabulary for rhymes. A generative n-gram model is used to generate sentences that are picked for the poem if they satisfy the requirements of word relatedness and rhyming. The results are evaluated with human judgement counting the number of thematically consistent words over the total number of words and open line transitions over all line transitions. This evaluation strategy relies on one person's opinion.

Twitter tweets can be combined into poems as suggested by the approach taken by Lamb et al. (2017). Their system groups tweets in their twitter corpus into categories based on their mutual rhyming. As they are focusing on sonnets, they remove all tweets that do not fill the syllabic criteria of a sonnet from the corpus as well as tweets that do not rhyme with any other tweet. Their system can rate tweets based on an emotion lexicon, trigram frequency and an imagery lexicon. They evaluated their system on a 5-point Likert scale using the following questions: *How much do you like this poem? How creative is this poem? How well does this poem express the emotion of [emotion]? How meaningfully does this poem summarize its topic? How new and different is this poem? How new and different is this poem? How cohesive is the narrative of this poem?*. The average values for each question were between 2 and 3.5. Interestingly, they did not perceive a significant difference between expert judges and non-expert judges.

A great deal of the previous work on poem generation has been mostly interested in superficial features, such as rhyme, meter and surface-level semantics, or just training a model to mimic poetry for the sake of it being trained on poems in hopes of the model learning the essence of poetry automatically. While our system in paper I does not extend far from the superficial, the poem generator in paper III has a more nuanced set of aesthetics that capture a wide range of characteristics typical to poems. Furthermore, neither of the systems is trained on poetry per se albeit they do use existing poems as templates. Especially important is the shift in the research focus in paper III from the output to the internal appreciation of the system.

3.3. The Master-Apprentice Approach

The practical foundation of the systems presented in paper II and III is on the master-apprentice framework. This means that there are two interacting systems in

place, a GA-based master and a seq2seq RNN-based apprentice. The work presented in paper I only consists of a single agent, a GA. The first version of the master-apprentice was developed by us in an earlier paper not included in this dissertation (Alnajjar & Hämäläinen, 2018). One advantage of the approach is that while the master can only generate a certain kind of output as defined by the evolutionary process, the apprentice can learn to extend from it by human authored data. In this way, the apprentice will not only mimic what humans have created, because part of the training comes from the master, neither will it only resonate the outcomes from our evolutionary algorithm as human authored data is also present.

3.3.1. Master as a Genetic Algorithm

In the papers, we model the master by using the same genetic algorithm implementation. The major differences between the papers are in the fitness functions and parameters for the genetic process. In addition, the poem generators in paper I and III differ in the initial population; the approach presented in paper I uses different poems in the population, whereas in III the whole population is initialized with the same poem. In this section, I will describe how the genetic algorithm operates in more detail.

We use a GA implementation provided in the DEAP library (Fortin et al., 2012). The algorithm is a standard $\mu + \lambda$ implementation. It operates on a population μ producing an offspring λ. During the process, mutations and crossovers will occur and the individuals in each generation are ranked based on the fitness functions using NSGA-II non-dominant sorting (Deb et al., 2002). NSGA-II is designed to rank individuals based on multiple parameters, as each individual is scored based on multiple metrics in the fitness function, ranking them becomes non-trivial.

In the very beginning, the algorithm needs an initial population. This population is based on existing artefacts. For movie title puns, it means that the initial population will be initialized with the movie title that is supposed to be converted into a punny form, for poetry generation, the population is initialized with existing poetry.

The population undergoes an evolutionary process for the desired number of generations. The evolutionary process has a great deal in common with the idea of modelling creativity as a search (c.f. Wiggins, 2006). The GA produces new individuals from existing ones by mutating them, the mutation has been implemented in a slightly different fashion in each paper, but in practice, the GA picks one word in an individual and replaces it with another one. Two individuals can also be selected for crossover, which in our papers means selecting one point in

the two individuals and swapping what follows after that point. The new individuals are added to the offspring produced by the current population.

At the end of each generation, a selection takes place. All individuals both in the current population and in the offspring are scored based on the fitness functions defined in each paper. The top fittest individuals are picked to survive to the next generation. It is important to note that the selection selects individuals from the current generation and the offspring. This ensures that if a good individual has been produced at some point during the evolutionary process, that individual is kept from generation to generation until better ones are produced. Otherwise, the quality of the individuals could degrade drastically over one generation if none of the new ones were on par with the previous ones.

For the Finnish language, the mutation does not merely replace words with others without any reference to grammatical knowledge as this would easily result in incomprehensible sentences. The system operates on lemmas that it will inflect to the morphology of the original context by using Omorfi (Pirinen, 2015) on UralicNLP (Hämäläinen, 2019). While this mostly solves the morphosyntactic requirements known as agreement, it does nothing to solve case government. For the case government, we use a practical NLG tool presented in paper VI.

For a better illustration of how the GA picks the best individuals for the next generation, I will describe some of the fitness functions used in the different papers. Please refer to papers I-III for full description of all the fitness functions used. The fitness functions range from the simple rule-based to more complex neural models. The simplest one that is common for both puns and poetry is the existence of rhyme in its various forms (full rhyme, consonance, assonance and alliteration). In poetry, the fitness functions measure the number of rhyming within the poem whereas for pun generation it is calculated between the original word and the potentially punny replacement word. For Finnish, we can determine rhyming easily with rules by looking into the characters of a word as the Finnish writing system is very phonetic.

For English, many existing approaches use CMU dict or ryhmes from Wiktionary to overcome the fact that the written form of words in English is not particularly phonetic. However, our initial master-apprentice approach (Alnajjar & Hämäläinen, 2018) needed something more robust as our system was dealing with words related to Saudi Arabia, some of which were direct loans from Arabic. This called for a more robust approach that can cover even atypical loan words. This is why our English rhyming mechanism uses a popular speech synthesizer eSpeakNG[2]

[2] https://github.com/espeak-ng/espeak-ng/

to convert English words into IPA. This makes it possible to determine rhyming with simple rules even with the difficulty resulting from the English writing system.

To capture sentiment in poetry, we use an already existing recent state of the art approach (Feng & Wan, 2019) as one of the fitness functions. The approach made it possible to train a sentiment analyzer for Finnish without annotated data in Finnish. Their method relies on training the model with English data and using bilingual word embeddings to predict sentiment for Finnish as well. The fitness function measures the variance of the sentiments within an individual poem, while preferring the values for variance learned from a poem corpus.

3.3.2. Apprentice as a Seq2seq Model

In our initial work on the master-apprentice approach (Alnajjar & Hämäläinen, 2018), there was only one master and one apprentice. The goal was to model computational creativity in the master so that it could produce creative movie title puns on its own. The apprentice was trained on the data produced by the master and human written movie title puns of the same domain, that is Saudi-Arabia related puns. The key idea behind having an apprentice is to permit the system to achieve the requirements for creative autonomy (see Jennings, 2010), that is, it can change its standards from what has been explicitly programmed into the master, while not fully relying on mere replication of human authored artefacts.

As the master operates on modifying an existing input (a movie title or a poem) and producing an output based on it, the task can equally be modelled as a sequence-to-sequence problem. The apprentice in papers II and III is a recurrent neural network (RNN) trained on a popular tool for training neural machine translation (NMT) models, known as OpenNMT (Klein et al, 2018).

The master-apprentice setting is extended in paper II to simulate different learning scenarios how the master teaches its apprentice. The scenarios were inspired by research on developmental psychology, more concretely we modelled the parenting styles identified by Baumrind (1991). The parenting styles can be divided into four categories, authoritarian, authoritative, rejecting-neglecting and permissive parenting, each of which was modelled as a different way the training of the apprentice took place.

Paper III extends this by introducing two different masters into the equation. This work does not rely on any psychological findings in human behavior but is more of a modest extension to how the master and apprentice have been utilised in the past. Furthermore, the difference this paper has regarding the previous work is that the apprentice is not trained on human authored data as such data is not present for the

task of Finnish poem generation by using existing poems as a starting point. In this work, creative autonomy is not approximated with the help of human authored data, but ultimately, by following two different masters.

4. From Theories to Practice

A definition of the problem one is set to solve is important when modelling anything computationally. Especially when the object of interest is something as elusive as creativity. For creativity can mean multiple different things to different people and without defining it before tackling it by computational means is like walking in a dark maze; disoriented, without any way of telling whether any of the steps taken have brought you anywhere nearer to the main goal.

In some more established fields of NLP, the problem definition is by no means needed to be explicitly stated, as they have established automated evaluation metrics. Therefore, the problem statement becomes more of "take any means necessary to get high scores on gold annotated data". This has led to a field that focuses more on the numerical truth of automated evaluation metrics than on advocating for a deep understanding of the real-life phenomenon being modelled. Nevertheless, automated evaluation metrics, however poor they might be (see for example Reiter, 2018; Talman & Chatzikyriakidis, 2019), make it possible to measure progress. It can now be said whether the steps taken in the dark maze have led closer to the exit or not.

Computational creativity research, it should be noted, typically does not explicitly define the problem that is being solved. Even if a definition is given, it is usually only for the sake of the technical implementation, and it is hardly given any reasoning as to why the particular definition was chosen and especially why it should model anything creative apart from purely generative. As a remark, Pollak et al. (2016) have found that the conceptualization of computational creativity, while exhibiting some stability, also changes over time, which highlights a part of the problem of researching creativity without defining it first.

I do like the ideology presented in the SPECS approach (Jordanous, 2012), and according to it, creativity must first be defined on an abstract level and then on a concrete level. I would also like to defend that any computational implementation of a creative system must try to address explicitly the requirements set in this definition; this is something overlooked by SPECS. Otherwise, the fact of having a definition loses its meaning. In the current era, it would be perfectly possible to come up with an elaborate definition of what poetry should be and just train a GAN model on raw poem data. If there is no alignment between the definition and the implementation, it becomes impossible to say to what degree the problem was

solved at all. It might very well be the case that the GAN just recycles solutions to the problem from human poets without creating anything of its own.

In the following sections, I will elaborate the theoretical foundations presented in paper II and III for humor and poem generation. Having an established definition of the problem to be solved makes it possible to track progress and evaluate the systems in a more meaningful way. Without any definition for creativity in poem generation, for example, any system capable of spitting out strings that resemble poetry would be sufficient.

4.1. Humor Generation

The abstract level definition for creativity in general was chosen to be that of the creative tripod, and more particularly focusing on the three legs of the tripod: skill, appreciation and imagination. In order to bring these notions to the context of humor generation, or pun generation, to be precise, it is important to look into what is known about puns and humor in a non-computational setting.

Many theories highlight that incongruity plays a crucial role and its resolution leads to a humorous effect (Oring, 2003; Attardo & Raskin, 1991). In practice, it means that two different scripts have to be possible at the same time, and they need to be in opposition (Raskin, 1985). However, a more concrete take on humor is that it requires surprise and coherence (Brownell et al., 1983), and we follow this definition in our approach. This theory does not completely deviate from the rest as surprise is close to the notion of incongruity, and coherence is what makes the resolution of surprise/incongruity possible.

The task itself of generating food related puns is not made up by us, but is based on a movement that took place on Instagram where people came up with food related puns out of existing movie titles, similarly to our system, such as *Harry Potter and the Deathly Marshmallows* instead of *Harry Potter and the Deathly Hallows*. These real-world puns authored by real people give us on the one hand, a narrowed down task to model, and on the other an additional source of training data for the apprentice.

We define that in order for the system to exhibit skill, it should be able to make a punny version out of an existing movie title in such a fashion that the original movie title still stays recognizable. These requirements for creativity are modelled in different parts of the system. The initial population of the master is initialized only with one original movie title to ensure the puns will be made for that specific title. One of the fitness functions measures the number of altered words in the movie title,

as the more words are altered, the more difficult it becomes to recognize the original movie title. Another fitness function is set to measure rhyming to ensure that the word replacements meet the minimum requirement of a pun, that is a sound similarity.

Now, for a pun to be funny, something more is needed. The appreciation the system has should be able to assess humor as it was defined based on existing theories. For this reason, fitness functions for evaluation of surprise and cohesion are implemented as they are needed for a joke to be funny. The fitness function evaluating sound similarity is part of the appreciation as well, as it not only tells that there is sound similarity, but also the degree to which it occurs.

The last requirement for creativity is imagination, which we define to be at least P-creativity in the output, but preferably H-creativity as well. This is taken care of by the genetic process that can find new punny titles novel to itself by mutation and crossover.

It is to be noted that only the implementation of the master is meant to reflect all the necessities for computational creativity in this particular context. The role of the apprentices is just a mere approximation of creative autonomy of the overall system when both the master and the apprentice are seen as a single whole. This is mostly since the apprentice is capable of learning both from the computationally creative master and human peers.

In relying on a theoretical foundation, we have made certain that there is a clear definition of what we are trying to solve, instead of just claiming to solve humor generation as a whole. Furthermore, it is possible to see a link between the theory and practice as what was being modelled corresponds to the individual parts of the definition. If a better way of solving any of the subcomponents of creative movie title pun generation emerged, it could be integrated to the current master presumably improving the quality of the output.

4.2. Poem Generation

In the context of paper III, creativity was first defined through the FACE model. In other words, the key components of creativity are now framing, aesthetics, concept and expression. Although we use these notions to define computationally creative poem generation, we do so only in the context of our poem generator. We do not seek to model poetry as a whole, as coming up with a definition capable of stretching over the whole genre of poetry is an endeavour too difficult even for those engaging in the practice of literary studies.

The expressions the system needs to produce are Finnish poems. Their creation is heavily guided by the aesthetics of the master. For the apprentice, they are produced by the sequence-to-sequence mappings it has learned from the input-output pairs of the master. All in all, for generating a new expression, an existing human authored poem is given as an input, and the expression output by the system is a new poem based on modification to the input.

On the level of concepts, there are masters and apprentices. Masters are programmed and designed by us, but they can adjust their appreciation based on a corpus. Apprentices are trained on the data of their masters. This means that there is a higher degree of flexibility in what these concepts are to become than what there is in the masters. Even more so, if they were to be exposed to human written parallel data.

The most crucial part for creativity lies in the aesthetics. These aesthetics are modelled as fitness functions in the master. The apprentice learns to mimic to a degree the qualities measured by these aesthetics in its output. The aesthetics followed are divided into four categories: sonic, semantic, imagerial, and metaphorical.

For the sonic aesthetics, the system must be able to appreciate the existence of rhyme, assonance, consonance or alliteration. In addition, the system should aim for maintaining the meter of the input poem. The meter is calculated by the number of syllables and the foot measured by the distribution of short and long syllables in each verse. This way, the system is not fixed to a certain meter for all poetry, but can flexibly use the meter that was in the input poem.

Semantics in poetry are different from those in everyday language and themes can vary within a poem drastically. This means that not all the words in a poem need to seem like they would fit into the same semantic fields. In fact, dividing words of a poem into their different semantic fields can reveal tensions and hidden interpretations of the poem (c.f. Lotman, 1974). Therefore, the system divides words into clusters by their semantic similarity by using affinity propagation (Frey & Dueck, 2007). The distances between the clusters are contrasted to the distances the master has learned from a corpus to know how distant two clusters must be and how distant they can be.

Imagery is an important aspect of poetry (see Kantokorpi et al., 1990), and in practice it refers to the mental images the act of reading a poem can evoke in its reader. In the philosophy of mind, these mental images are known as qualia (see Chalmers, 1995), and they are indeed considered a hard problem of mentality, far beyond the reach of a mere machine. However, the system should be able to evaluate

them with its aesthetics. For this reason, the master implements a fitness function for sentiment analysis in order to get a better grasp of the potential sentiment evoked in the reader. Also, as Kao & Jurafsky (2012) point out echoing the words of Burroway (2007), concrete words can be used to provoke imagery. For this reason, the master also measures the concreteness of the words of a poem to assess its imagery.

The final aspect of metaphors in poetry is assessed by finding metaphorical relations between the semantic clusters of a poem. As poems might have vocabulary that does not form a semantically meaningful whole, they are likely to make sense pragmatically, i.e. through interpretation of figurative language such as the metaphor. Metaphors can be understood as consisting of a tenor and a vehicle (Richards, 1936). The metaphoricity of having the centroid of one cluster as a tenor and the centroid of another one as the vehicle is assessed by automatic measures adapted from Alnajjar (2019).

Framing is used in our approach to make the aesthetics used by the master for its poetry more transparent. As it is easy to retrieve the individual values from the different fitness functions for the generated poetry, it is possible to provide a framing by filling a template with slots for the framing information to be provided. This framing is then used when conducting human evaluation, as discussed in the following sections.

5. On Evaluation of Computationally Creative Systems

As we have seen in the related work section, there are many ways of conducting evaluation of systems aiming for creative language generation. Some papers only present automated evaluation, while some conduct a human evaluation. Whatever the evaluation metric, it is usually picked in an ad-hoc manner (see Lamb et al., 2018).

The SPECS approach does state that evaluation should come from the definition of creativity, an idea I could not agree more with. Needless to say, I cannot emphasize enough how important it is that the implementation of the system also try to explicitly model what has been defined. Otherwise one would not be evaluating the effectiveness of the proposed computational solution, but rather whether people are ready to perceive something in the output that the system was not aware of itself. Thus, any merit that the output of the system might have, does not come from the system itself but is due to serendipity, to a mere chance or the characteristics of the corpus used in training.

In fact, Veale (2016) points out that people are ready to interpret a deeper meaning in computer generated text, providing that it has a suitable linguistic form. Such is the case for any system that does not try to model any of the evaluated metrics in particular. Quite many papers, as we saw in the related work section, evaluate poem generation based on metrics such as poeticity or meaningfulness, while the actual implementation focuses merely on enforcing rhyme and meter in a model trained on existing poems. This is cumbersome especially since poems are meant to be interpreted. A seemingly irrational combination of words might gain a deeper meaning when read by humans.

In the next sections, I present the evaluation and my findings from paper I. The evaluation in this paper follows exactly an ad-hoc approach that is more or less the typical way of going about evaluation in CC and creative NLG. In paper V, I show the bigger problems arising from this type of evaluation. I shall also describe the evaluation from paper II that aimed for evaluation based on a model that implements the requirements defined for movie title pun generation. And finally, I will describe

the results of paper III, where the aim was also to reduce the possibility of interpretation and people reading more into the output, when evaluating the system.

I am not taking a stance on the question of using expert evaluators versus amateur evaluators. While some findings suggest that there is no difference (Lamb et al., 2017), while others suggest that experts are more consistent (Toral et al., 2018). However, I am afraid this question might be very specific to the task that is being solved and to the solution proposed. Furthermore, experts might introduce another source of bias especially if they have a strong opinion on how a particular kind of art should be, rather than what it could be.

5.1. Ad-Hoc and Abstract

In this section, I will describe the evaluation presented in paper I, and more importantly I will discuss the findings presented in paper V on the same evaluation metric. This section focuses on evaluation with abstract ad-hoc questions.

What I mean by ad-hoc evaluation questions is that the evaluation seems to come from nothing. Usually there is no reasoning behind the evaluation questions or they are taken as such from existing research. Typically they do not try to address the different aspects that are being modelled or are considered important for the problem definition. Such an evaluation can be found to a great extent in contemporary papers dealing with creative NLG, and paper I is no exception.

In this way, it is possible to gather evaluation results that look convincing but do not really tell anything about the system. The evaluation questions used in paper I were, *How typical is the text as a poem? How understandable is it? How good is the language? Does the text evoke mental images? Does the text evoke emotions? How much do you like the text?,* evaluated on a 5-point Likert scale and one binary question *Is the text a poem?*. These questions are very typical ones and they have been used in previous research as well. The problem arises that not unlike in many existing papers out there, the fitness functions of the system only measured different kinds of rhyming, poetic foot and rudimentary semantics. In other words, the system neither attempted, nor was it capable of optimizing for typicality, mental images, emotions or general likability. Out of the evaluation questions, it only aimed for comprehensibility as semantics had been considered during the creative process and grammaticality as the methods presented in paper VI were in place.

Everything else is mainly about the evaluator's ability to read more into the poem than what the system ever intended. This is indeed a problem, especially in the era of neural networks, when it is now easier than ever to train a model to produce good

sounding text. The only thing that does matter to evaluation questions of this nature is whether people are ready to read more into the output of the system, to project a deeper meaning to the poem.

I would like to highlight that the initial idea of SPECS is not enough, meaning that evaluation should follow from a definition, but the implementation must also conform to the initial definition. It would not be impossible for me to come up with a definition for creativity for paper I based on the evaluation questions to make the definition and evaluation be in line. But it would still not change the fact that the implementation does not even try to solve a majority of the attributes that are evaluated. Unfortunately this evaluation practice is omnipresent in the field and it makes it next to impossible to compare the different systems proposed over the years.

Abstractness of the evaluation questions is another thing that I have learned to be a matter to avoid. One of the times I was conducting poem evaluation on these very questions, one of the evaluators struggled on the first question, he looked at me with agony on his face and said: "How can you ask something as difficult as this? Is the text a poem? And yet, you are giving me a text without its context, for it is the context that makes a poem a poem!" This moment was revealing, the more abstract the questions, the more room for interpretation there is. Even the question whether something is a poem or not is open to interpretation, to a high degree of subjectivity. And all this would have gone unnoticed had I conducted the evaluation online on FigureEight or Amazon Mechanical Turk. Removing the outliers, as suggested by some, can hardly solve this problem. For example, the person who struggled with the first question answered no differently to the questions than his peers. It is to be noted that people can take very different interpretative strategies in answering the abstract evaluation questions, but these cannot necessarily be seen in the data collected.

I have evaluated my first poem generator (Hämäläinen, 2018) that used rules, knowledge bases and a statistical language model to generate poetry with the same evaluation questions. In the same paper, I also evaluated a method that took full sentences out of a corpus and put them together to form a poem. Both of these methods were able to get better results on most of the metrics in human evaluation than what an existing system had received when it had been evaluated. This perhaps, highlights the problem even further, that even combining poems out of existing sentences with no creative intent, can indeed yield state-of-the-art results.

In paper V, I generated poems with my first poem generator (Hämäläinen, 2018) and converted them into 3 different Finnish dialects automatically. The evaluation was conducted with the same abstract questions, but this time, my intention was not

to showcase a new system with claimed computationally creative capabilities, but to see how something as superficial as a dialect can affect the evaluation results. This time, the poems that were not adapted to any dialect got worse results than in my initial evaluation presented in the original paper. This points towards the fact that there is not much stability in using abstract questions over time. One important factor might be that in 2018, none of the evaluators even thought that computers could be used to generate poems, and they felt surprised that they had read poetry with no human author, whereas in 2020, many of the evaluators asked after the study whether the poems were computer generated.

Our findings on dialects suggest that even though dialectically adapted poems scored lower on all the metrics than the original non-dialectal poem generated by the system, there was an observable tendency. The further away the dialect was from the written standard, the lower it scored. This can perhaps point towards a familiarity bias meaning that people are more likely to prefer things familiar to them than those they are unfamiliar with. Based on these evaluation results, we can say that dialect that is usually an uncontrolled variable can affect the results, though negatively. However, dialect can also have a positive effect.

Paper V presents another evaluation, where we asked people to associate words with either the original standard written Finnish poem or its dialectal counterpart. Words *creative* and *original* were considerably more frequently and *poem-like* slightly more frequently associated with the dialectically adapted poem, whereas *fluent* was considerably more frequently associated with the standard written Finnish poem and *emotive* and *artificial* slightly more frequently. This means that if this kind of an association type evaluation was followed, the uncontrolled variable of dialect can make poems to be perceived as more creative and original.

5.2. Theory-based and Abstract

In paper II, I present my work on moving away from ad-hoc evaluation questions into something more reasoned by the actual approach taken. This involves theoretical definition, a computationally creative system that implements the definition and evaluation questions that are directly derived from the definition. As we will see in this section, this is but a step towards a more meaningful evaluation, but the abstract nature of the evaluation questions continues to pose challenges yet to be tackled.

As described in section *4.1. Humor generation*, the master was tailored to work on the creative tripod framework consisting of skill, imagination and appreciation.

For skill, we used the following evaluation statements: *The title has a pun in it, The title is related to food* and *The original title is recognizable*. For appreciation the statements were: *The title is humorous, The pun is surprising* and *The pun makes sense in the context of the original movie*. And finally, the statements for imagination were *The pun in the title is obvious* and *The pun in the title sounds familiar*. The evaluation was conducted on a 5-point Likert scale, higher values indicating more agreement with the statement.

The evaluation statements are measuring the individual aspects that we defined important for the task and implemented in the master. For instance, the statements for appreciation ask about humor and its subcomponents separately, as it was the subcomponents, surprise and coherence, that were explicitly modelled in the system. In fact, in our previous approach (Alnajjar & Hämäläinen, 2018) we found this evaluation sufficient. However, the aspect of having multiple apprentices in paper II quickly revealed the shortcomings of the evaluation.

We ran the same evaluation on a crowdsourcing platform for movie title puns generated by the master and four apprentices in different learning scenarios: authoritarian, authoritative, rejecting-neglecting and permissive. The different learning scenarios can be shortly described so that in the authoritarian scenario, the apprentice only got training data from the master, in the authoritative one, the training data came from the master and from peer data filtered by the fitness functions. In the rejecting-neglecting scenario, the apprentice only learned from peer data and in the permissive one, the apprentice learned from all the data.

Our initial impression when looking at the results by ourselves was that permissive and rejecting-neglecting produced the worst results. This was also supported by our automated evaluation metric that was based on the fitness functions. When looking at the results from the human evaluation, however, this conclusion was no longer as straight forward. In fact, the permissive apprentice got the best scores on recognizability of the original title, making sense in the context of the original movie title, not being obvious and not sounding familiar. None of these metrics result in a high level of humor or punniness, however. Neglecting scored the highest on food relatedness, humor and surprise.

The authoritarian apprentice scored the highest only on punniness and surprise. However, it consistently scores high on all the metrics, as does the neglecting apprentice. As none of the apprentices nor the master is the single best one on all the metrics, it becomes difficult to suggest any model over another. Especially since the authoritative apprentice got the clearly highest score on punniness, which was the

ultimate goal of the approach, but scored lower than the neglecting apprentice on the other metrics important for creativity and humor.

The evaluation questions are still abstract and leave room for interpretation. When taking a closer look at the evaluation results for individual annotators, I could see many cases where, for instance, titles that did not have a pun in them in my opinion were rated as punny and vice versa for titles that did have a clear pun. Thus subjectivity and possibility for interpretation are still present with these evaluation questions. Having test questions is not a solution either as they might introduce bias in the selection of human judges. If the judgments of only those who answer "correctly" to test questions are considered, rather than improving the quality of the evaluation, it might ensure that only judges like-minded with us are selected. This would indeed yield the desired results, but do so without any scientific validity.

It is also important to note that the rejecting-neglecting apprentice was only trained with puns authored by humans. Yet it scored high on our evaluation metrics that were supposed to measure computational creativity, not a mere copying behavior. On one hand, this supports my claim that training a model to generate based on data will deliver good results, but it does not reach creativity (at least not by the definition we followed in the paper), on the other hand it highlights a problem in the evaluation metrics and underlines how important it is that the computational model is also explicitly aware of the necessary metrics. All in all, even though this evaluation is a step in the right direction, it still suffers from the abstractness of the evaluation questions and that a non-creative system can still score high on the metrics.

5.3. Theory-based and Concrete

The work presented in paper III continues the ideology of theory driven evaluation as established earlier. For poem generation, a different theoretical approach, namely FACE was chosen. The theory defines creativity through framing, aesthetics, concepts and expressions. The main reason behind leaving the creative tripod behind and changing the theoretical framework was to see what might follow from a different theory in terms of computational modelling and evaluation. As it turned out, framing was an important aspect for solving abstractness in the evaluation questions.

In this section, I will discuss the framing and aesthetics evaluation presented in the paper as I consider it the most meaningful out of the evaluations. The evaluation

of concepts was conducted automatically and the evaluation of expressions was a matter of preference between the master and the apprentice.

We used the notion of framing to fill a template of evaluation questions with the data provided by the master's fitness functions. These evaluation questions/ statements serve as a framing for the system to explain its creative decisions. The framing used for evaluation consisted of the following questions: *Do the words written in italics have rhymes (e.g. heikko peikko)?, Do the words written in italics have assonance (e.g. t**a**l**o** s**a**n**o**)?, Do the words written in italics have consonance (e.g. s**a**kk**o** s**o**kk**a**)?* and *Does the poem have alliteration within a verse (e.g. **v**anha **v**esi)?*. And the following statements: *Verse number X and Y have the same meter, The poem has X semantic fields: [semantic cluster 1]... and [semantic cluster N], The semantic fields [semantic cluster X] and [semantic cluster Y] are the closest to each other, The semantic fields [semantic cluster A] and [semantic cluster B] are the furthest away from each other, The following words in the poem [concrete words] are concrete concepts, The verse number X is positive, The verse number Y is negative, The following words in the poem [metaphorical words] can be understood metaphorically* and *The word X has a metaphorical connection to word Y*. The system would mark all rhyming words of any kind with italics and fill in the placeholders in the evaluation statements with the fitness functions.

As a difference to the previous papers, we did not use a 5-point Likert scale in this evaluation. Instead, we gave people three options: agree, disagree and I don't know. The reason behind this is simple, it is easier to interpret what it means when people either agree or disagree than if people showed an average agreement of 3.25 on one system versus 3.31 on another, which was one of the problems of the evaluation presented in paper II. Also, this binary scaling removes some of the subjectivity that might arise from people using a Likert-scale. Having an option I don't know is also meant to reduce bias in the results. People are not forced to give an opinion, if they cannot clearly form one, which means that they are less likely to just pick one of the two options at mere random when they find the assessment difficult.

Again, the evaluation is targeted at the exact features that were modelled computationally in the fitness functions. But now, we are forcing the evaluation to concretely evaluate the output of the fitness functions. If, for example a metaphor was predicted, we do not merely ask if the poem has a metaphor, which would be very open to interpretation; people might even understand something metaphorically the system was not aware of during the creative process. But we give a listing of the predicted metaphorical words and even an example of a metaphorical interpretation

by the machine. This reduces the high level of subjectivity typically involved in CC evaluation even further by giving less room for interpretation.

The results obtained this way are more revealing of the shortcoming of the system than in the previously presented papers I and II. Even though rhyming is measured by rule-based metrics that work well for a language like Finnish that has a highly phonetic writing system, none of the rhyming questions received a 100% agreement by people and the fitness functions. In fact, we can indeed find a shortcoming in the fitness functions based on this evaluation. While they do measure the presence of the different types of rhyming correctly, they do not measure their quality. This means, for instance, that words such as *en* (I don't) and *et* (you don't) would be rated as having assonance by the system as they fill the main criterion: two different words that share the same vowels but have different consonants. However, such an assonance is hardly perceivable by people. This indicates that in the future, the system should measure rhyming by some metric more nuanced than just its mere existence in the poem. Consonance rhyme scored particularly lowly, which is mostly due to the fact that it is not the most typical type of rhyming in Finnish poetry and people are unfamiliar with it.

The statements relating to semantic clusters received the highest number of I don't know answers from the judges, this was also the case for the last statement about metaphors. This result, even though it does not really help in gaining knowledge about the performance of the method on these aspects, is still useful. It highlights that the questions of this kind, while being very important for poetry, are still difficult for people to assess when asked this concretely. This raises the question what the abstract evaluation questions that are typically used really measure, how can we feel any confidence about the results of a questionnaire assessing metaphoricity of a poem in a 5-point Likert scale using an abstract question, when people tend not to know when the question is asked more concretely? The results for the penultimate evaluation statement of metaphoricity showed relatively high agreement and lower number of I don't know answers. However, this statement leaves more room for interpretation than the last one.

For sentiment analysis, the method used clearly excelled in predicting positive sentiment, but did not work nearly as well for negative sentiment. Perhaps this is due to the fact that sentiment in poetry is conveyed and understood very differently than in regular language. For instance, *autumn leaves are falling,* is neither negative nor positive on a surface level, yet in a poem it might convey the idea of death or an end of an era of joy and happiness.

All in all, we can say that this evaluation method shows more clearly the problems of the system and gives some exact ideas for improvement such as the quality of rhymes and a sentiment analysis that caters more to sentiment in poetry in particular. As for the statements receiving a high number of I don't know answers, a more qualitative approach could be taken in the future in evaluating them. Or perhaps their nature is such that these questions can better be assessed by experts in poetry than ordinary people.

6. Conclusions and Future Work

The contributions of this thesis reach to all of the three important components of computational creativity: theory, practice and evaluation. Paying attention to all of the three instead of only one alone has made it possible for symbiosis between the three components, which I believe to be vital for any work aiming to build computationally creative systems. Theoretical definitions are needed to narrow down and properly describe the problem one seeks to solve by computational means; they are important for motivating creativity in the proposed practical solution and ultimately for formulating the evaluation of the system. Without a definition for the problem one seeks to model, it is impossible to tell the degree to which the problem has been solved by the proposed solution. This type of practice without clear definitions will doom the field to not advancing from a stage where new models can be proposed easily, but their intercomparison is difficult and progress is not measurable.

From the theoretical perspective, in this thesis, I have presented a definition for generating movie title puns and Finnish poetry, both of them are based on different higher-level theories of computational creativity. It is important to note that the resulting definitions are not the only possible ones, and especially, in the case of poetry, the definition was not meant to define the ever-changing genre of poetry as a whole, but rather it was to narrow it down to something that can be computationally modelled.

The larger theoretical contribution of my thesis was the elaboration of a theoretical framework of my own. My theory, unlike the existing ones on computational creativity, explicitly states the need for meaning. People do not use language exclusively to evoke a certain emotional response or exhibit artistic value by their wording. Words are used to convey a concrete meaning, and this should also be considered in future research aiming to generate creative language. However, this is a difficult task indeed and the practical applications of this thesis have merely focused on the aesthetic rather than the communicative. This would, however, be an interesting line of work which I will be heading towards in the future.

From the practical point of view to computational creativity, we have elaborated a master-apprentice approach. The genetic algorithm, the master, serves an important

role in making it possible to model the individually important aspects of computational creativity as defined from a theoretical point of view. A genetic algorithm also makes it possible to have diversity in the output due to the randomness introduced by the genetic process itself. This is particularly useful and desirable behaviour in a creative system. The NMT model, the apprentice, makes it possible to approximate autonomous creativity, and it gives us some intriguing information how human authored data affects the results when perceived by people.

The tale of the master and the apprentice is far from over. Currently, we are interested in taking the approach more towards the direction of a proper multi-agent system. This opens up a great deal of new research questions one can study in such a setting. Such as how would learning from very distinctive masters affect the results of the apprentice, or what if masters could learn from apprentices or apprentices from one another. The approach could be also tried out in contexts outside of the computational creativity domain to see if using a genetic algorithm to generate training data in a resource poor scenario has a positive impact when training an LSTM model.

Evaluation has been of particular interest throughout the work presented in this thesis. Evaluation should be conducted in such a way that it evaluates the individual aspects that have been computationally modelled. A system scoring high on a metric it was not aware of can hardly do so because of its own merit, but rather because people are willing to read more into its output than what the system ever intended. Furthermore, the evaluation becomes more useful in pointing out the merits and shortcomings of the system if it is done with as concrete evaluation questions as possible. Abstract questions leave too much room for interpretation making it more difficult to say whether the measured feature was the one the system intended or if the evaluators read more into the output.

Nevertheless, evaluation is a question far from solved despite the findings presented in this thesis. It is still an open question the degree to which one should rely on amateurs and experts in the evaluation process. The evaluation process itself is also something that should be more carefully studied in the future. In our evaluations, we have always shuffled the order of the artefacts so that there would be as little a constant priming effect as possible and we have been careful in not revealing to the judges they are evaluating computer generated artefacts. However, not all papers conduct the evaluation in this manner. In the future, it would be important to see whether shuffling or not and revealing that the artefacts are generated by a computer or not has a real measurable effect on the evaluation results.

During the process of my doctoral research, I have also published several open-source Python libraries relating to NLG that people can use freely. These are UralicNLP[3], Syntaxmaker[4] (described in paper VI), FinMeter[5] (the aesthetics of paper III) and Murre[6] (described in paper V). All of these are also permanently archived on Zenodo with a new DOI for every new release.

[3] https://github.com/mikahama/uralicNLP
[4] https://github.com/mikahama/syntaxmaker
[5] https://github.com/mikahama/finmeter
[6] https://github.com/mikahama/murre

References

Aggarwal, S., & Mamidi, R. (2017). Automatic generation of jokes in Hindi. In *Proceedings of ACL 2017, Student Research Workshop* (pp. 69-74).

Alnajjar, K (2019). Computational Analysis and Generation of Slogans. Master's Thesis. *University of Helsinki, Faculty of Science*

Alnajjar, K., & Hämäläinen, M. (2018). A master-apprentice approach to automatic creation of culturally satirical movie titles. In *Proceedings of the 11th International Conference on Natural Language Generation* (pp. 274-283).

Asmis, E. (1992). Plato on poetic creativity. In R. Kraut (Ed.), *The Cambridge Companion to Plato* (Cambridge Companions to Philosophy, pp. 338-364). Cambridge: Cambridge University Press. doi:10.1017/CCOL0521430186.01

Attardo, S., & Raskin, V. (1991). Script theory revis (it) ed: Joke similarity and joke representation model. *Humor-International Journal of Humor Research*, 4(3-4), 293-348.

Baumrind, D. (1991). Parenting styles and adolescent development. *The Encyclopedia of Adolescence* (pp. 758–772).

Bedworth, J., & Norwood, J. (1999). The Turing test is dead.... In P*roceedings of the 3rd Conference on Creativity & Cognition* (pp. 193-194).

Bertero, D., & Fung, P. (2016). Deep learning of audio and language features for humor prediction. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)* (pp. 496-501).

Boden, M. A. (2004). The creative mind: Myths and mechanisms. Routledge.

Brownell, H. H., Michel, D., Powelson, J., & Gardner, H. (1983). Surprise but not coherence: Sensitivity to verbal humor in right-hemisphere patients. *Brain and language*, 18(1), 20-27.

Burroway, J. (2007). Imaginative writing: The elements of craft. *Longman Pub Group*.

Caccavale, F., & Søgaard, A. (2019). Predicting Concrete and Abstract Entities in Modern Poetry. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 33, pp. 858-864).

Chalmers, D. J. (1995). Absent qualia, fading qualia, dancing qualia. *Conscious experience*, 309-328.

Chen, P. Y., & Soo, V. W. (2018). Humor recognition using deep learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)* (pp. 113-117).

Chikai, K., Takayama, J., & Arase, Y. (2019). Responsive and Self-Expressive Dialogue Generation. In *Proceedings of the First Workshop on NLP for Conversational AI* (pp. 139-149).

Colton, S. (2008). Creativity Versus the Perception of Creativity in Computational Systems. In *AAAI spring symposium: creative intelligent systems* (Vol. 8).

Colton, S., Charnley, J. W., & Pease, A. (2011). Computational Creativity Theory: The FACE and IDEA Descriptive Models. In *The 2nd International Conference on Computational Creativity* (pp. 90-95).

Colton, S., & Wiggins, G. A. (2012). Computational creativity: the final frontier?. In *Proceedings of the 20th European Conference on Artificial Intelligence* (pp. 21-26).

Cook, M., Colton, S., Pease, A., & Llano, M. T. (2019). Framing in computational creativity–a survey and taxonomy. In *Proceedings of the 10th International Conference on Computational Creativity* (pp. 156-163).

Davis, N. M. (2013). Human-computer co-creativity: Blending human and computational creativity. In *Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*.

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2), 182-197.

Feng, Y., & Wan, X. (2019). Learning Bilingual Sentiment-Specific Word Embeddings without Cross-lingual Supervision. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (pp. 420-429).

Fortin, F. A., Rainville, F. M. D., Gardner, M. A., Parizeau, M., & Gagné, C. (2012). DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 13(Jul), (pp. 2171-2175).

Frey, B. J., & Dueck, D. (2007). Clustering by passing messages between data points. *Science*, 315(5814), 972-976.

Gaut, B. (2012). Creativity and Rationality. *The Journal of Aesthetics and Art Criticism*, 70(3), (pp. 259-270). www.jstor.org/stable/43496511

Gervás, P. (2018). Targeted Storyfying: Creating Stories About Particular Events. In *the Proceedings of the 9th International Conference on Computational Creativity* (pp. 232-239).

Gervás, P. (2017). Template-Free Construction of Poems with Thematic Cohesion and Enjambment. In *Proceedings of the Workshop on Computational Creativity in Natural Language Generation (CC-NLG 2017)* (pp. 21-28).

Goffman, E. (1959) The Presentation of Self in Everyday Life. *University of Edinburgh Social Sciences Research Centre*

Grice, H. P. (1975). Logic and conversation. In *Speech acts* (pp. 41-58). Brill.

He, H., Peng, N., & Liang, P. (2019). Pun Generation with Surprise. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (pp. 1734-1744).

Hennessey, B. A., & Amabile, T. (2010) Creativity. *Annual Review of Psychology*, 61(1), (pp. 569-598).

Honkela, T. (2017). Rauhankone: tekoälytutkijan testamentti. *Gaudeamus*.

Hossain, N., Krumm, J., & Gamon, M. (2019). "President Vows to Cut< Taxes> Hair": Dataset and Analysis of Creative Text Editing for Humorous Headlines. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (pp. 133-142).

Hämäläinen, M. (2018). Harnessing NLG to Create Finnish Poetry Automatically. In the Proceedings of the 9th International Conference on Computational Creativity (pp. 9-15). Association for Computational Creativity (ACC).

Hämäläinen, M. (2019). UralicNLP: An NLP library for Uralic languages. *Journal of Open Source Software*, 4(37), 1345.

Hämäläinen, M., & Alnajjar, K. (2019). Creative contextual dialog adaptation in an open world RPG. In *Proceedings of the 14th International Conference on the Foundations of Digital Games*.

Jennings, K. E. (2010). Developing creativity: Artificial barriers in artificial intelligence. *Minds and Machines*, 20(4), (pp. 489-501).

Jordanous, A. (2012). A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative. *Cognitive Computation*, 4(3), (pp. 246-279).

Kantokorpi, M., Viikari, A., & Lyytikäinen, P. (1990). Runousopin perusteet. *Gaudeamus*.

Kao, J., & Jurafsky, D. (2012). A computational analysis of style, affect, and imagery in contemporary poetry. In *Proceedings of the NAACL-HLT 2012 workshop on computational linguistics for literature* (pp. 8-17).

Kim, J. (2018). Philosophy of mind. *Routledge*.

Klein, G., Kim, Y., Deng, Y., Nguyen, V., Senellart, J., & Rush, A. M. (2018). OpenNMT: Neural Machine Translation Toolkit. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)* (pp. 177-184).

Lau, J. H., Cohn, T., Baldwin, T., Brooke, J., & Hammond, A. (2018). Deep-speare: A joint neural model of poetic language, meter and rhyme. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1948-1958).

Lamb, C., Brown, D. G., & Clarke, C. L. (2018). Evaluating computational creativity: An interdisciplinary tutorial. *ACM Computing Surveys (CSUR)*, 51(2), (pp. 1-34).

Lamb, C., Brown, D. G., & Clarke, C. L. (2017). Incorporating novelty, meaning, reaction and craft into computational poetry: a negative experimental result. In *Proceedings of 8th International Conference on Computational Creativity* (pp. 183-188).

Loller-Andersen, M., & Gambäck, B. (2018). Deep Learning-based Poetry Generation Given Visual Input. In *the Proceedings of the 9th International Conference on Computational Creativity* (pp. 240-247).

Lotman, J. M., (1974). Den poetiska texten. *Stockholm*.

Lubart, T. (2005). How can computers be partners in the creative process: classification and commentary on the special issue. *International Journal of Human-Computer Studies*, 63(4-5), (pp. 365-369).

Luo, F., Li, S., Yang, P., Li, L., Chang, B., Sui, Z., & Xu, S. U. N. (2019, November). Pun-GAN: Generative Adversarial Network for Pun Generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 3379-3384).

Manurung, R., Ritchie, G., & Thompson, H. (2012). Using genetic algorithms to create meaningful poetic text. *Journal of Experimental & Theoretical Artificial Intelligence*, 24(1), 43-64.

Marsella, S., Gratch, J., & Petta, P. (2010). Computational models of emotion. *A Blueprint for Affective Computing-A sourcebook and manual*, 11(1), 21-46.

McCharty, J. (2007) What is artificial intelligence? Technical report, Computer Science Department, Stanford University

Gonçalo Oliveira, H., (2017). A survey on intelligent poetry generation: Languages, features, techniques, reutilisation and evaluation. In *Proceedings of the 10th International Conference on Natural Language Generation* (pp. 11-20).

Gonçalo Oliveira, H., & Alves, A. O. (2016). Poetry from concept maps–yet another adaptation of PoeTryMe's flexible architecture. In *Proceedings of 7th International Conference on Computational Creativity, ICCC.* (pp. 246-253)

Gonçalo Oliveira, H., Costa, D., & Pinto, A. M. (2016). One does not simply produce funny memes! – explorations on the automatic generation of internet humor. In *Proceedings of the Seventh International Conference on Computational Creativity (ICCC 2016).* (pp. 238-245).

Oring, E. (2003). Engaging humor. *University of Illinois Press*.

Pease, A., & Colton, S. (2011). On impact and evaluation in computational creativity: A discussion of the Turing test and an alternative proposal. In Proceedings of the AISB symposium on AI and Philosophy (Vol. 39).

Pirinen, T. A. (2015). Development and Use of Computational Morphology of Finnish in the Open Source and Open Science Era: Notes on Experiences with Omorfi Development. SKY Journal of Linguistics, 28. (pp. 381-393).

Pollak, S., Boshkoska, B. M., Miljkovic, D., Wiggins, G. A., & Lavrac, N. (2016). Computational creativity conceptualisation grounded on ICCC papers. In *Proceedings of the Seventh International Conference on Computational Creativity*. (pp. 123-130)

Rahgozar, A., & Inkpen, D. (2019). Semantics and Homothetic Clustering of Hafez Poetry. In *Proceedings of the 3rd Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature* (pp. 82-90).

Raskin, V. (1985). Semantic mechanisms of humor (Vol. 24). *Springer Science & Business Media*.

Reiter, E. (2018). A Structured Review of the Validity of BLEU. *Computational Linguistics*, 44(3), 393-401.

Rhodes, M. (1961). An analysis of creativity. *The Phi Delta Kappan*, 42(7), (pp. 305-310).

Richards, I. A. (1936). The philosophy of rhetoric. *Oxford University Press*

Searle, J. R. (1969). Speech acts: An essay in the philosophy of language (Vol. 626). *Cambridge university press*.

Shao Y, Zhang C, Zhou J, Gu T, Yuan Y. (2019) How Does Culture Shape Creativity? A Mini-Review. *Front Psychol*. 2019;10:1219. Published 2019 May 28. doi:10.3389/fpsyg.2019.01219

Shen X, Su H, Li Y, Li W, Niu S, Zhao Y, Aizawa A, Long G. A. (2017). Conditional Variational Framework for Dialog Generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 504-509).

van Stegeren, J., & Theune, M. (2019). Churnalist: Fictional Headline Generation for Context-appropriate Flavor Text. In *10th International Conference on Computational Creativity* (pp. 65-72). Association for Computational Creativity.

Talman, A., & Chatzikyriakidis, S. (2019). Testing the Generalization Power of Neural Network Models across NLI Benchmarks. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP* (pp. 85-94).

Toral, A., Castilho, S., Hu, K., & Way, A. (2018). Attaining the Unattainable? Reassessing Claims of Human Parity in Neural Machine Translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers* (pp. 113-123).

Veale, T. (2018) A Massive Sarcastic Robot: What a Great Idea! Two Approaches to the Computational Generation of Irony. In *the Proceedings of the 9th International Conference on Computational Creativity*. (pp. 120-127).

Veale, T. (2016). The shape of tweets to come: Automating language play in social networks. *Multiple Perspectives on Language Play*, 1, 73-92.

Veale, T. & Alnajjar, K. (2016) Grounded for life: creative symbol-grounding for lexical invention, *Connection Science*, 28:2, (pp. 139-154), DOI: 10.1080/09540091.2015.1130025

Wei, W., Le, Q., Dai, A., & Li, J. (2018). Airdialogue: An environment for goal-oriented dialogue research. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 3844-3854).

Wiggins, G. A. (2006). A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems,* 19(7), (pp. 449-458).

Winters, T., Nys, V., & De Schreye, D. (2019). Towards a general framework for humor generation from rated examples. In *Proceedings of the 10th International Conference on Computational Creativity* (pp. 274-281).

Yang, D., Lavie, A., Dyer, C., & Hovy, E. (2015). Humor recognition and humor anchor extraction. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (pp. 2367-2376).

Yang, Z., Cai, P., Feng, Y., Li, F., Feng, W., Chiu, E. S. Y., & Yu, H. (2019). Generating Classical Chinese Poems from Vernacular Chinese. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (pp. 6156-6165).

Yannakakis, G. N., Liapis, A., & Alexopoulos, C. (2014). Mixed-initiative co-creativity. In P*roceedings of the 9th Conference on the Foundations of Digital Games*.

Yi, X., Sun, M., Li, R., & Li, W. (2018). Automatic poetry generation with mutual reinforcement learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (pp. 3143-3153).

Yu, Z., Tan, J., & Wan, X. (2018). A neural approach to pun generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1650-1660).

Zhang, J., Feng, Y., Wang, D., Wang, Y., Abel, A., Zhang, S., & Zhang, A. (2017). Flexible and Creative Chinese Poetry Generation Using Neural Memory. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1364-1373).